

固定制御可検査性に基づくRTL回路の非スキャンテスト容易化設計法

永井 慎太郎, 和田 弘樹, 大竹 哲史, 藤原 秀雄

奈良先端科学技術大学院大学 情報科学研究科
〒630-0101 奈良県生駒市高山町 8916-5
Tel: 0743-72-5226 Fax: 0743-72-5229

E-mail: {shinta-n, hiroki-w, ohtake, fujiwara}@is.aist-nara.ac.jp

あらまし 本稿では、レジスタ転送レベル回路を対象としたテスト容易化設計法を提案する。対象回路は、コントローラ、データバスおよびそれらを接続する内部接続線で構成される。提案手法では、内部接続線を疑似外部入出力として、コントローラとデータバスそれぞれに対してテスト容易化設計およびテスト生成を行う。コントローラに対するテスト容易化設計では、既発表の我々のテスト容易化設計法を用いる。データバスに対するテスト生成は、階層テスト生成法に基づき、回路中の各組合せ回路要素に対してテストプランを生成する。また、データバスのテスト容易化設計では固定制御可検査性を導入する。固定制御可検査性とは、各組合せ回路要素に対するテストプランでの正当化/テスト/伝搬の3つのフェーズにおける制御ベクトル系列を1個の制御ベクトルで構成可能な回路の性質をいい、その性質に基づくデータバスのテスト容易化設計法を提案する。これにより、データバスに対して付加するテストプラン生成回路の構成を組合せ回路で実現できる。また実験結果により提案手法の有効性を示す。

キーワード レジスタ転送レベル回路, 階層テスト生成法, テストプラン, 固定制御可検査性

A Non-Scan DFT Method for RTL Circuits Based on Fixed-Control Testability

Shintaro Nagai, Hiroki Wada, Satoshi Ohtake and Hideo Fujiwara

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara 630-0101
Tel: 0743-72-5226 Fax: 0743-72-5229

E-mail: {shinta-n, hiroki-w, ohtake, fujiwara}@is.aist-nara.ac.jp

Abstract In this paper, we present a non-scan DFT method with complete fault efficiency for RTL circuits. An RTL circuit generally consists of a controller and a data path. The controller and the data path are connected with internal signals: control signals and status signals. We apply non-scan DFTs in the controller and in the data path, respectively. For the controller, we apply our DFT method which add an invalid state test generator to the controller. For the data path, test generation is based on hierarchical test generation. We introduce a new testability called "fixed-control testability" in order to generate a test plan for every combinational module. A test plan consist of three phases: justification, test and propagation phase. For fixed-control testability, each phase of test plan is composed of only one vector. Therefore, the test plan generator for the data path can be designed with combinational logic. Our experimental result show that the proposed method can reduce significantly both of test generation time and test application time compared with the full-scan design, though the hardware overhead of our method is slightly as same as that of the full-scan design.

Key words RTL circuit, hierarchical test generation, test plan, fixed-control testability

1 はじめに

近年のVLSI回路の大規模化・高集積化に伴い、VLSI回路に対するテストはますます困難な問題となっており、テストの費用削減およびテストの質の向上が求められている。テストの費用はテスト生成やテスト実行にかかる時間で評価できる。また、テストの質は、故障検出効率で評価できる。故障検出効率とは、回路中のテスト生成の対象となる全故障数に対する、テスト生成アルゴリズムによって生成されたテスト系列が検出可能な故障数とテスト生成アルゴリズムが冗長と判定した故障数の和との割合をいう。特に故障検出効率が100%の場合を完全故障検出効率という。

組合せ回路に対しては、効率の良いテスト生成アルゴリズムが提案されており、実用的なテスト生成時間で完全故障検出効率を持つテスト系列を求めることができる[1]。これに対して、順序回路では、一般に実用的なテスト生成時間で高い故障検出効率を持つテスト系列を求めることは困難である。そのため、順序回路をテスト生成の容易な回路に設計変更するテスト容易化設計法が提案されている。

代表的なテスト容易化設計法には完全スキャン設計法[1]がある。完全スキャン設計法では、順序回路中の全てのフリップフロップ(FF)をスキャンFFに置き換え、FFの値を外部から直接制御および観測を可能にする。これにより、テスト生成の際に順序回路中の全てのFFを擬似的に外部入出力とみなすことができるので、組合せ回路部分に対して組合せ回路用のテスト生成アルゴリズムが適用可能となり、短いテスト生成時間で完全故障検出効率を持つテストベクトル集合を生成することができる。しかし、この手法には、比較的大きい面積オーバーヘッドおよび長いテスト実行時間を要し、実動作速度でのテスト(at-speed test)[2]が困難で、回路規模が増大するとテスト生成時間が増大するなどの問題点がある。

完全スキャン設計法での問題点を解消するために、種々の非スキャンテスト容易化設計法が提案されている。さらに、ゲートレベルでの回路要素数は膨大であるためゲートレベルでのテスト生成およびテスト容易化設計は非常に困難となるので、近年、レジスタ転送(RT)レベルでのコントローラ/データバス回路を対象とするテスト生成やテスト容易化設計法が提案されている。

RTレベルでのコントローラのテスト容易化設計法として、我々が提案したコントローラの非スキャンテスト容易化設計法[3]がある。これは、無効テスト状態生成器と呼ばれる回路をコントローラに対して付加することにより、組合せ回路部分に対して組合せ回路用のテスト生成アルゴリズムが適用でき、短いテスト生成時間で、完全故障検出効率を持つテスト系列を生成することができる。

RTレベルでのデータバスのテスト容易化設計法として、Genesis[5]や強可検査テスト容易化設計法[6]などがある。これらの手法は階層テスト生成法[4]に基づいている。階層テスト生成法は次の2段階から成る。第1段階ではゲートレベルにおいて、組合せ回路で構成される演算器やマルチプレクサなどの回路要素単体に対して組合せ回路用のテスト生成アルゴリズムを用いてテスト生成を行う。第2段階ではRTレベルにおいて、回路要素毎のテストプランを生成する。テストプランとは、外部入力からその回路要素の入力ベクトルを伝搬(正当化)し、その回路要素の出力応答を外部出力へ伝搬するための、制御ベクトルの時系列をいう。テストプランは、テストベクトルの正当化/テスト/出力応答の伝搬の3つのフェーズで構成され、一般にテストプランでの各フェーズにおける制御ベクトルは、時刻毎に変化する。

RTレベルでのコントローラ/データバス回路全体に対する非スキャンテスト容易化設計法として、コントローラおよびデータバスそれぞれに対して文献[3]および[6]を適用した手法[7]を提案した。この手法では、小さい面積オーバーヘッドで、テスト生成時間およびテスト実行

時間を完全スキャン設計法に比べて大幅に短縮している。しかし、データバスに対してテストプランを供給するためのテストプラン生成回路(TPG)は順序回路で構成されるので、その面積オーバーヘッドが問題であった。

本稿では、RTレベルでのコントローラ/データバス回路全体に対するテスト容易化設計法を提案する。コントローラに対するテスト容易化設計では、文献[3]を適用する。データバスに対するテスト容易化設計では、TPGの面積オーバーヘッドを縮小するためにデータバスの新しいテスト容易性として固定制御可検査性を提案する。固定制御可検査性とは、各回路要素に対するテストプランにおいて各フェーズの制御ベクトル系列が各々1個の制御ベクトルで構成できるデータバスの回路構造の性質をいう。固定制御可検査性を有するデータバスでは、回路要素毎のテストプランは高々3個の制御ベクトルで構成できるので、テストプランのデータバスへの供給が文献[7]に比べて容易となり、さらにTPGの構成が組合せ回路で実現できるので、文献[7]に比べてTPGの面積オーバーヘッドを縮小できる。本稿ではさらに、与えられたデータバスを固定制御可検査性を満たすデータバスに設計変更するためのテスト容易化設計法およびテストプラン生成法を提案する。ベンチマーク回路および実設計回路を用いた実験では、提案手法は文献[7]でのテスト生成時間およびテスト実行時間と同等である。データバスの面積オーバーヘッドは、文献[7]と同等であり、TPGおよびコントローラ/データバス全体の面積オーバーヘッドは文献[7]に比べて縮小できたことを示す。

以下、2節では本稿で対象となるデータバスおよびコントローラを定義する。3節ではRTレベルでのコントローラ/データバス回路全体に対するテスト容易化設計法について提案する。さらに、コントローラに対するテスト容易化設計法[3]を紹介し、固定制御可検査性に基づくデータバスのテスト容易化設計法およびテストプラン生成法を提案する。4節ではベンチマーク回路および実設計回路を用いた実験により、提案手法の有効性を示す。

2 諸定義

2.1 コントローラとデータバス

RTレベルでのVLSI回路は一般に、コントローラとデータバスから成る(図1)。コントローラは状態遷移図で記述され、データバスは回路要素と回路要素を接続する信号線で記述される。回路要素には、外部入力、外部出力、ホールド機能を持つレジスタと持たないレジスタ、マルチプレクサ(MUX)、加算器や乗算器などの演算モジュールおよび比較器などの観測モジュールに分類される。また各回路要素はポートを持つ。ポートはデータポート、制御ポートおよび観測ポートに分類される。データポートは、回路要素にデータを入力する入力ポートと回路要素からデータを出力する出力ポートに分類される。本稿では、全ての回路要素のデータポートのビット幅は等しいものとする(ビット幅の異なる場合への拡張は可能である)。制御ポートは、回路要素を制御するための信号をコントローラから入力するためのポートである。観測ポートは、回路要素のステータスをコントローラへ出力するためのポートである。

信号線は、データ信号線、制御信号線およびステータス信号線に分類される。データ信号線は、各回路要素のデータポートに接続され、2つの回路要素を接続する。制御信号線は、コントローラからデータバスの各回路要素の制御ポートに制御信号を与えるための信号線である。ステータス信号線は、データバスの各回路要素の観測ポートからコントローラへステータス信号を出力するための信号線である。

以下、MUX、演算モジュールおよび観測モジュールなどの組合せ回路で構成される回路要素を組合せ回路要素と呼ぶ。

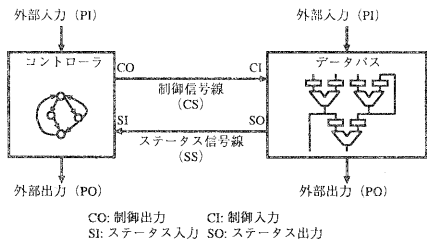


図 1: RT レベル回路例

2.2 階層テスト生成法

以下では、データバスの RT レベル記述とデータバス中の組合せ回路要素のゲートレベル記述が与えられるものとする。

階層テスト生成法は、次の 2 段階から成る。与えられたデータバスに対して、第 1 段階ではゲートレベルにおいて、各組合せ回路要素単体に対して組合せ回路用のテスト生成アルゴリズムを用いてテストベクトル集合を生成する。第 2 段階では RT レベルにおいて、各組合せ回路要素に対してテストプラン生成を行う。テストプランは、外部入力から対応する組合せ回路要素へのテストベクトルの正当化、およびその出力応答の外部出力への伝搬を行うための制御ベクトルの時系列をいう。一般に組合せ回路要素 M のテストプランは正当化フェーズ、テストフェーズ、伝搬フェーズで構成される。ここで、 M の直前のレジスタの集合を R_J 、直後のレジスタの集合を R_P とする。

正当化フェーズ: 正当化フェーズでは、外部入力から $r_j \in R_J$ の入力ポートにテストベクトルを正当化する。 M の全ての入力ポートに対して、 R_J が空の場合は、正当化フェーズは存在しない。

テストフェーズ: テストフェーズは M の入出力ポート、制御ポートおよび観測ポートについて以下のように分類する。 M の全ての入力ポートに対して、 R_J が空でない場合は、各 $r_j \in R_J$ から M にテストベクトルを与える。 M の入力ポートのうち、ある入力ポートに対する直前のレジスタがない場合は、外部入力から直接テストベクトルを与える。 M の全ての入力ポートに対して R_J が空の場合は、外部入力から M に直接テストベクトルを与える。 M の出力ポートに対して、 R_P が空でない場合は、各 $r_p \in R_P$ に出力応答を伝搬する。 R_P が空の場合は、外部出力に直接出力応答を伝搬する。制御ポートに関して、 M が制御ポートを持つ場合には、外部から制御ポートへテストベクトルを与える。観測ポートに関して、 M が観測ポートを持つ場合には、観測ポートに現れた出力応答を外部から観測する。

伝搬フェーズ: 伝搬フェーズは r_p から外部出力に出力応答を伝搬する。 R_P が空の場合、伝搬フェーズは存在しない。

2.3 データバスの強可検査性

RT レベルでのデータバスに対する非スキャンテスト容易化設計法として、文献 [6] がある。本節では、文献 [6] で提案したデータバスの強可検査性を定義する。

定義 1 (強可検査性 [6]) データバス DP 中の各組合せ回路要素 M に対してテストプランが存在し、そのテストプランで外部入力から M の入力ポートへ任意の値を正当化、かつ M の出力ポートから任意の値を外部出力へ伝搬できるとき、 DP は強可検査であるという。 □

強可検査性を満たす DP には、次の利点がある。

- 各組合せ回路要素単体に対して、組合せ回路用のテスト生成アルゴリズムを適用するので、完全故障検出効率を持つテストベクトル集合を短いテスト生成時間で求めることができる。

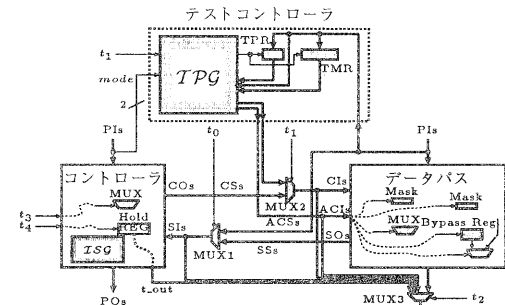


図 2: テストコントローラを含む RT レベル回路

- テストプラン生成は RT レベルで生成されるため、テストプラン生成時間が短い。
- 各組合せ回路要素に対して、外部入力から任意の値を正当化でき、その任意の出力応答を外部出力へ伝搬できるテストプランの生成を保証しているため、 DP 全体に対して、完全故障検出効率を達成できる。

3 RT レベルコントローラ/データバス回路のテスト容易化設計法

3.1 概要

文献 [7] の手法では、与えられた RT レベルでのコントローラ/データバス回路 (図 1) のコントローラおよびデータバスそれぞれに対して文献 [3](3.2 節) および文献 [6] を適用する。文献 [3], [6] の手法では、コントローラ/データバス回路の制御信号線およびステータス信号線は外部から直接制御/観測が可能であると仮定している。この仮定を満たすために、コントローラとデータバス間にテスト用の回路を付加する。コントローラのテストにおいては、ステータス信号線の外部からの制御/制御信号線の外部での観測を可能とする必要がある。そこで、データバスの外部入出力を通して、これらの信号線が直接制御、観測できるように MUX を挿入する (図 2)。データバスのテストにおいては、制御信号線を通してテストプランを供給する。コントローラの外部入力のビット幅は、一般に制御信号線のビット幅よりも小さいので、外部から直接テストプランの供給を行おうとすると、外部ピンオーバヘッドが大きくなる。そこで、テストプランをデータバスへ供給するための回路 (テストプラン生成回路) をコントローラ/データバス回路内部に付加する。ステータス信号線の観測は、外部で直接観測できる必要がある。そこで、ステータス信号線については、データバスの外部出力を介して直接観測する。文献 [7] では、各組合せ回路要素に対するテストプランを構成する制御ベクトルが時刻毎に変化するため、テストプラン生成回路は順序回路で構成され、その面積が大きくなる傾向がある。そこで本稿では、小さな回路で生成できるテストプランが存在するためのデータバスの性質として、固定制御可検査性を定義する。本稿で提案する RT レベルでのコントローラ/データバス回路のテスト容易化設計法では、コントローラに対しては文献 [3] を適用し、データバスに対しては固定制御可検査性に基づくテスト容易化設計法 (3.3 節) を適用する。コントローラとデータバス間には文献 [7] に準ずる回路を付加するが、テストコントローラについては固定制御可検査性に即したテストプラン生成回路を用いる。以下では、コントローラおよびデータバスのテスト容易化設計法、テストコントローラの構成について述べる。

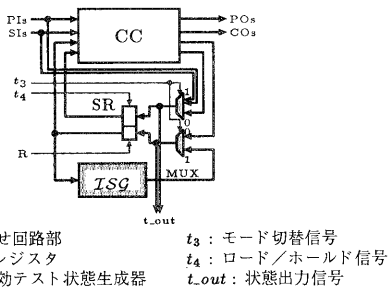


図 3: ISG を含むコントローラ

表 1: テストピンの制御

テストピン					Operation
t_0	t_1	t_2	t_3	t_4	
0	0	0	0	0	通常動作
1	0	1	*	*	コントローラのテスト
0	1	*	X	X	データバスのテスト

*: テストボタンまたはテストプランに依存する
X: ドントケア

3.2 コントローラのテスト容易化設計法

本節では、状態遷移図から合成されたコントローラに対するテスト容易化設計法 [3] の概要を述べる。この手法では、コントローラのテスト生成に対して、短いテスト生成時間で完全故障検出効率を達成するテストベクトル集合を生成でき、また at-speed テストが可能である。この手法では、順序回路の組合せ回路部 CC に対して組合せ回路用のテスト生成アルゴリズムを用いてテストベクトル集合を生成する。組合せ回路部 CC に対するテストベクトルは、外部入力 PI、ステータス信号線および状態レジスタ SR に対応する値で構成される。テストベクトルの CC への印加において、テストベクトルの SR に対応する値を SR に設定する必要がある。その値が状態遷移図のリセット状態から到達可能であれば、通常動作の状態遷移を用いてその値を SR に設定できる。リセット状態から到達不可能な場合は、無効テスト状態生成器 (ISG) と呼ばれる付加回路をコントローラに付加することで、その値を SR に設定する (図 3)。図 3 において、 t_3 は CC の出力または ISG の出力のどちらかを SR にロードするかを選択するために利用する。 t_4 はテスト実行時間を削減するための SR のロード・ホールド機能を実現するために利用する。

ここで、ステータス信号線の可制御性および可観測性、制御信号線の可観測性を保証するために、MUX1 および MUX3 を付加する (図 2)。ステータス信号線のビット幅は、一般にデータバスの外部入力のビット幅に比べて小さいので、これらの MUX の付加によりステータス信号線の制御が可能となる。一般にデータバスの外部出力のビット幅は制御信号線のビット幅に比べて大きいので、制御信号線の観測はデータバスの外部出力から観測する。 t_3 、 t_4 およびこれらの MUX の制御 t_0 、 t_2 は表 1 のように行う。

3.3 データバスのテスト容易化設計法

本節では、データバスの固定制御可検査性を定義する。与えられたデータバスを固定制御可検査性を満たすためのテスト容易化設計アルゴリズムを提案する。さらに、固定制御可検査性を有するデータバス (固定制御可検査データバス) 上の各組合せ回路要素に対するテストプラン生成アルゴリズムを提案する。

3.3.1 固定制御可検査性

定義 2 (固定制御可検査性) データバス DP が以下の条件を満たすとき、 DP は固定制御可検査であるという。

C1: DP は強可検査である。

C2: DP 中の各組合せ回路要素に対するテストプランにおける正当化/テスト/伝搬の各フェーズを構成する制御ベクトル系列が各々 1 個の制御ベクトルで構成される。□

固定制御可検査データバスは、条件 C1 を満たすので、強可検査データバスの利点を持つ。さらに固定制御可検査データバスでは、各組合せ回路要素に対するテストプランの制御ベクトル数が各々 3 であるため、テストプラン生成回路を文献 [7] に比べて容易に構成できる。

3.3.2 テスト容易化設計アルゴリズム

提案するテスト容易化設計法では、与えられたデータバス中の各組合せ回路要素に対して、任意のテストベクトルを外部入力から正当化するための経路 (制御経路)、および任意の出力応答を外部出力へ伝搬するための経路 (観測経路) を決定する。次に各組合せ回路要素に対して決定した制御経路および観測経路を用いて任意の値を伝搬できることを保証するためのこれらの経路上でいくつかのテスト用の回路を付加する。制御経路および観測経路上で以下 2 点を実現できれば、これらの経路上で任意の値の伝搬が保証できる。

- 演算モジュールの入出力ポート間に存在する経路上での任意の値の伝搬
- 順序深度の等しい再収束経路上での値の伝搬のタイミング衝突の解消

ここで、順序深度とは経路上に存在するレジスタの個数をいう。再収束経路とは異なる 2 つの回路要素 e_1 および e_2 に対して、 e_1 を始点とし、 e_2 を終点とする任意の異なる経路の対のうち、 e_1 、 e_2 以外に共通部分を含まないものをいう。

提案手法では、演算モジュールの入力ポートから出力ポートへの任意の値の伝搬を保証するために、スルー機能を入出力ポート間に付加する。スルー機能は、回路要素の 1 個の入力ポートに印加した任意の値を出力ポートへ伝搬させるための機能である。また、順序深度の等しい再収束経路上のタイミング衝突を解消するために、MUX、またはバイパスレジスタを再収束経路の一方の経路上に挿入して、再収束経路を構成する一方の経路上の順序深度を 1 減らすか、または 1 増やす。

以下は、与えられたデータバスを固定制御可検査データバスに設計変更するのに十分なテスト容易化設計アルゴリズムを提案する。

Step1 制御林の生成

与えられたデータバス DP (図 4(a)) 内の各組合せ回路要素に対して制御経路を決定する。一般には、ある外部入力からある入力ポートへの経路は複数存在するが、テスト実行時間を短縮するためには、順序深度が最小の経路を選択することが望ましい。またスルー機能が付加される場所を極小化するためには、各組合せ回路要素に対する制御経路同士ができるだけ多くの共有部分を持つことが望ましい。そこで、各組合せ回路要素に対する制御経路の集合を DP 上のレジスタ数に関する最小構成林として生成する (図 4(b))。ここで、組合せ回路要素の入力ポートについて、制御経路の終点となるものを非伝搬入力、終点でないものを伝搬入力と呼ぶ。図 4(b) において、加算器の入力ポート x_1 は非伝搬入力、 y_1 は伝搬入力となる。

Step2 観測林の生成

DP 内の各組合せ回路要素に対して観測経路を決定する。一般には、ある出力ポートからある外部出力への経路は複数存在するが、制御経路との共有部分が極大であるようなものを選択することにより、観測経路上で任意の値を伝搬させるために付加されるスルー機能の数を削

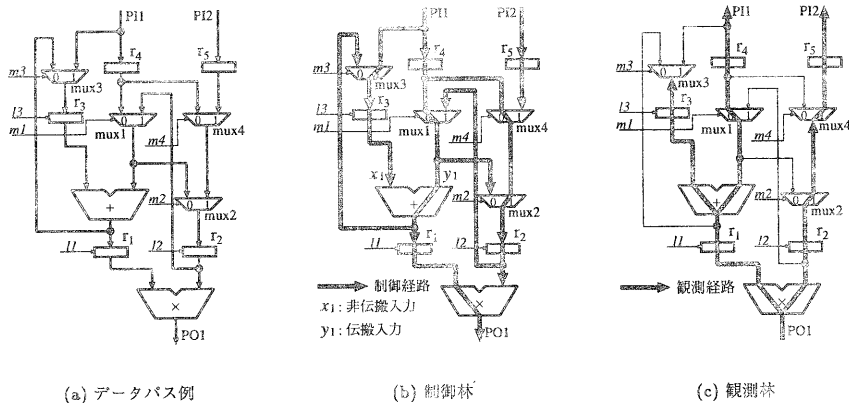


図 4: データパス例とその制御林および観測林

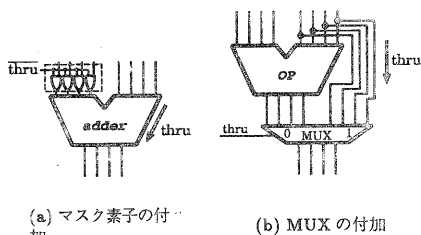


図 5: スルー機能の実現例

減することが可能となる。従って、各組合せ回路要素に対する観測経路から成る集合を DP の外部出力を根とする DP 上の非伝搬入力に関する最小構成林として生成する (図 4(c))。

Step3 スルー機能の付加 (制御林上の値の伝搬の保証)

Step1 で生成された全ての制御経路上で任意の値を伝搬できることを保証する。データパス中の各回路要素に対する入出力ポート間での値の伝搬について考える。レジスタの入出力間においては、ロード信号とクロック信号を与えることにより任意の値を伝搬できる。MUX の入出力間においては、制御信号を与えることにより任意の値をスルーできる。演算モジュールの伝搬入力と出力ポート間にスルー機能がある場合は、スルー命令を実行すれば、入出力間の値の伝搬を保証できる。スルー機能がない場合は、演算モジュールの非伝搬入力と出力ポート間にスルー機能を付加する。スルー機能は、加算器、乗算器などの一般的な演算モジュールであれば、適切なマスク素子を非伝搬入力に付加することでスルーを低面積で実現できる (図 5(a))。マスク素子は、演算モジュール内の伝搬入力と出力ポート間で任意の値を伝搬するのに必要な定数を非伝搬入力に対して発生する。マスク素子を利用してスルー機能を実現できない場合は、MUX を付加してスルー機能を実現する (図 5(b))。これにより、 DP 中の全ての制御経路上で任意の値が伝搬できる。

Step4 MUX またはバイパスレジスタの付加 (順序深度の等しい再収束経路上のタイミング調整)

Step1 で生成した制御経路上で、2つの入力ポートを持つ組合せ回路要素 M に対して、ある回路要素を始点とする順序深度の等しい再収束経路が構成される場合がある。ここで、 M の各入力ポートに対して異なる値を伝搬することを考える。順序深度の等しい再収束経路上においてレジスタが存在する場合は、一般にレジスタのホールド機能を用いて異なる値を伝搬できるが、固定制御の条件のもとではレジスタのホールド機能は利用できない。提

案手法では、MUX またはバイパスレジスタをこのような再収束経路上に付加して任意の異なる値の伝搬を保証する。一般に2入力ポートを持つ複数の組合せ回路要素に対して、順序深度の等しい再収束経路が構成される場合がある。この場合、これら全ての再収束経路について付加回路を挿入すると、付加回路による面積オーバーヘッドが大きくなると考えられる。そこで提案手法では、制御経路を上流にある (外部入力から近い) 2入力ポートを持つ組合せ回路要素からタイミング調整を行い、下流にある組合せ回路要素については、上流で付加された回路を再利用することを考える。これにより、付加回路の面積オーバーヘッドを縮小できると期待できる。ここで、再収束経路の始点および終点となる回路要素をそれぞれ e_1 および e_2 とする。再収束経路を構成する制御経路で、 e_1 の出力ポートから e_2 の伝搬入力に接続された制御経路を p_c とする。また、 p_c 上のレジスタの集合を R とするとき、 R について次の2つの場合に分けて設計変更を行う。

場合 1: 集合 $R \neq \phi$

p_c 上にレジスタが存在するので、MUX を付加してレジスタをバイパスすることで、MUX を付加した経路上の順序深度を1つ減らすことができる。付加した MUX を TM とし、 TM の付加方法を以下に示す。

e_1 の直後のレジスタを r とする。 r の出力ポートを z' とし、 r の直後の回路要素の伝搬入力を x' とする。また、 TM の2つの入力ポートを x および y とし、出力ポートを z とする。このとき、 TM は r の直後に付加する。これは、 r の直後に分岐が存在する場合に分岐幹に TM を付加することで、付加した TM の共有が可能となる場合がある。これにより、付加回路の面積オーバーヘッドを縮小できると期待できる。 TM の入出力ポート x と z は、次のように接続する。 z' と x' を接続する信号線を除去し、 z' と x 、 z と x' を接続する。ここで、伝搬入力に接続された制御経路上に TM を付加するのは、 TM 自身のテストを考慮しているため、観測経路を選ぶ際に伝搬入力を含むように観測経路を構成しているので、 TM の出力応答の伝搬のための観測経路を新たに張り直す必要はない。

次に、 TM の入力ポート y の接続を考える。 r が自己ループを持つ場合、 r の直前の信号線と y を接続すると、非同期ループが発生する。そこで、非同期ループの発生を防ぐための接続方法を考える。外部入力から r までの制御経路上にあるレジスタの集合を R_i とする。このとき、 R_i について次の2つの場合に分けて TM の入力ポート y の接続を決定する。

(a) $R_i \neq \phi$ の場合

r に最も近いレジスタ r' を集合 R_i から選択し、 r' の出力ポートと TM の入力ポート y を接続する。

(b) $R_i = \phi$ の場合

外部入力と TM の入力ポート y を接続する。

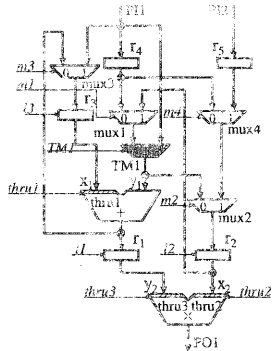


図 6: 固定制御可検査データパス

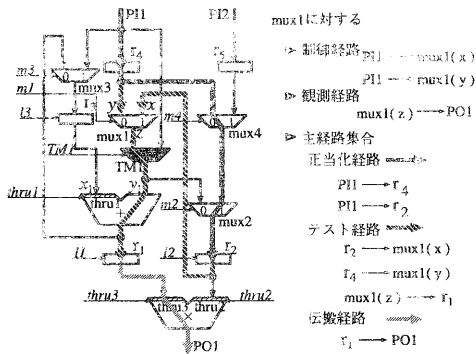


図 7: 主経路集合 (図 6:mux1)

ここで、 TM の入力ポートへの制御経路は再収束経路となるが、順序深度が異なることが保証できるので、 TM のテストのための制御経路を新たに構成する必要はない。場合 2: 集合 $R = \phi$

p_a 上にレジスタが存在しないので、バイパスレジスタを用いて、付加した経路上のレジスタの段数を 1 増やすことにより、順序深度の等しい再収束経路上のタイミング調整を行う。一般に、 e_1 の出力ポートと e_2 の伝搬入力間に分岐が存在する場合がありますので、 e_1 の直後にバイパスレジスタを付加する。これにより、付加したバイパスレジスタの共有が可能となる場合があり、付加回路の面積オーバーヘッドを縮小できると期待できる。ここで、バイパスレジスタを構成する MUX のテストについて考える。場合 1 と同様に、付加したバイパスレジスタの出力ポートは e_2 の伝搬入力に接続しているため、バイパスレジスタを構成する MUX の出力応答の伝搬のための観測経路を新たに張り直す必要はない。また、バイパスレジスタ中の MUX の入力ポートへの経路は再収束経路となるが、順序深度が異なることが保証できるので、この MUX のテストのための制御経路を新たに構成する必要はない。

以下では、制御経路上に付加した MUX またはバイパスレジスタがある場合には、制御経路を変更することなく、MUX またはバイパスレジスタを付加した分だけその経路上の順序深度を増減できるものとする。

Step5 スルー機能の付加 (観測経路上の値の伝搬の保証)

Step2 で生成した観測経路が 2 入力演算モジュール M を通る場合を考える。観測経路が M の伝搬入力上を通る場合、Step3 により M の伝搬入力と出力ポート間のスルーができるので任意の値の伝搬は保証できる。観測経路が M の非伝搬入力上を通る場合、 M の非伝搬入力と出力ポート間のスルー機能を保証していない。ここで、伝搬入力に定数を与えて非伝搬入力と出力ポート間のス

時刻	PI1	PI2	m1	m2	m3	m4	TM1	i1	i2	i3	thru1	thru2	thru3	PO1	フェーズ
1	C	X	X	X	X	X	X	X	X	X	X	X	X	-	正当化フェーズ
2	C	X	X	1	X	0	X	1	X	X	X	X	X	-	正当化フェーズ
3	X	X	T	X	X	0	1	X	X	X	X	X	X	-	テストフェーズ
4	X	X	X	X	X	X	X	X	X	X	X	X	X	1	伝搬フェーズ

C: 外部入力へのテストパターン印加タイミング X: ドントケア
 O: 外部出力での出力応答の観測タイミング
 T: 制御ポートへのテストパターンの印加タイミング

図 8: テストプラン例 (図 6:mux1)

表 2: テストコントローラの制御

t_1	mode	Operation
0	0 0	$TPR, TMR \leftarrow PI$ リセット
1	0 1	$Z \leftarrow TP_p (TMR)$ 正当化
	1 0	$Z \leftarrow TP_t (TMR, TPR)$ テスト
1	1 1	$Z \leftarrow TP_b (TMR)$ 伝搬

TP_p : 正当化フェーズの制御ベクトル t_1 : テストピン
 TP_t : テストフェーズの制御ベクトル mode: モード切替信号
 TP_b : 伝搬フェーズの制御ベクトル Z: 制御信号線

ルー機能を実現できる場合について考える。外部入力からその入力ポートへ定数を印加できれば、スルー機能を新たに付加する必要はなくなる。この場合、スルー機能実現のための付加回路面積を削減することができる。そこで、外部入力から定数を M の伝搬入力へ印加するための経路 (補助経路) の利用について考える。

M の非伝搬入力を x 、伝搬入力を y とする。また M の出力ポートを z とする。ここで、 y の制御経路を p_y 、 x の制御経路を p_x とする。以下では、 x の制御経路の始点となる外部入力を $ctrlPI(x)$ 、外部入力から x の制御経路の順序深度の集合を $depth(x)$ とする。

経路 p_x 上に存在する組合せ回路要素の集合を C_M とする。 $c_M \in C_M$ の入力ポートを x_M および y_M とし、 $depth(x_M) = a$ とする。Step4 で DP に付加した MUX の個数を m 、バイパスレジスタの個数を l とすると、外部入力から x_M までの制御経路の順序深度の集合は、 $a - m \leq depth(x_M) \leq a + l$ である。

補助経路が利用可能な場合は 2 通りあり、(1) 順序深度に関係なく $ctrlPI(x_M) \cap ctrlPI(y_M) \neq ctrlPI(y)$ の場合、(2) $ctrlPI(x_M) \cap ctrlPI(y_M) = ctrlPI(y)$ であり、かつ $depth(x_M) \cap depth(y_M) \neq depth(y)$ の場合である。補助経路が利用不可能な場合は、 $ctrlPI(x_M) \cap ctrlPI(y_M) = ctrlPI(y)$ であり、かつ $depth(x_M) \cap depth(y_M) = depth(y)$ の場合である。この場合は、 M の伝搬入力と出力ポート間にスルー機能を付加する。

例: 図 4 に対して固定制御可検査テスト容易化設計アルゴリズムを適用した結果を図 6 に示す。Step3 では、加算器および乗算器それぞれに対してスルー機能 thru1 および thru2 を付加する。これにより Step1 で生成した制御経路上の値の伝搬を保証できる。図 4 において加算器、乗算器および mux2 が順序深度の等しい再収束経路の終点となる回路要素である。Step4 では、はじめに加算器に対して $TM1$ を付加する。次に mux2 に対してテスト容易化設計を適用するが、 $TM1$ を付加しているのに対して新たに DFT 要素を付加する必要はない。乗算器に対しても同様である。これらの再収束経路の終点となる回路要素のテストには $TM1$ を用いる。Step5 では、加算器および乗算器の p_x 上に存在する回路要素の出力応答を伝搬する際に補助経路を利用可能かどうか調べる。加算器に対して、 $depth(y_1) = \{0, 1\}$ であり、 $depth(x_1) = \{1, 2\}$ であるので、補助経路を利用できる。乗算器に対して、 $depth(y_2) = \{1, 2\}$ であり、 $depth(x_2) = \{1, 2\}$ であるので、補助経路を利用できない。そこで、乗算器の伝搬入力と出力ポート間にスルー機能 thru3 を付加する。

3.3.3 テストプラン生成アルゴリズム

本節では、固定制御可検査データパス上の各組合せ回路要素に対してテストプランを生成するアルゴリズムを

表 3: コントローラ/データバス回路の特性

circuit	Area(#gate)	コントローラ					データバス						
		#PI	#PO	#State	#Status	#Control	Area(#gate)	#PI	#PO	#bit	#Reg.	#Mod.	Area(#gate)
GCD	1524.50	0	1	4	3	7	169.40	32	16	16	3	1	1350.90
JWF	6875.40	0	0	8	0	38	199.50	80	80	16	14	3	6671.70
LWF	1986.20	0	0	4	0	8	57.70	32	32	16	5	3	1924.30
PAULIN	24965.60	0	0	6	0	16	123.50	64	64	32	7	4	24833.70
RISC	62287.60	0	2	11	54	62	3986.90	32	96	32	40	4	58157.90

述べる。テストプラン生成アルゴリズムの入力は、固定制御可検査データバスの制御林および観測林である。

主テストと副テスト 2つの入力ポート (x, y) , 出力ポート z を持つ組合せ回路要素 M において, M を通るループが存在する場合, M の一方の入力ポート x に対する制御経路 cp 上に M が現れることがある。この場合, M のテスト実行の際に, $y-z$ 間を通して x にテストベクトルを伝搬する必要があるが, M に故障がある場合は, $y-z$ 間でテストベクトルを正しく伝搬できない可能性がある。そのため, $y-z$ 間で正しくテストベクトルを伝搬できるかどうかを調べる必要がある。そこで, M に対する各テストベクトルが外部入力から cp を通って外部出力まで正しく伝搬するかを調べる副テストを行う。 M の副テストで故障が検出されなければ, M に対するテストベクトルを cp 上を伝搬させて, M をテストする。これを主テストという。副テスト用のテストプランは, 主テスト用のテストプランと同様に生成できるので, 以下では主テスト用のテストプラン生成のみについて述べる。

主経路集合の生成 テストプランにおける各フェーズの制御ベクトルを求めるためには, 各フェーズの経路上に含まれる回路要素の中で制御ポートを持つ回路要素 M_i を求める必要がある。そこで, 正当化経路, テスト経路および伝搬経路を決定する。また, 補助経路を利用するにはこれも決定する。以下は, これら4つの経路の集合(主経路集合)を生成する手順である。

テスト対象の組合せ回路要素 M_i を考える。 M_i に対する制御経路および観測経路を基に M_i のテストプランを生成する。制御経路上で M_i の入力ポートの直前のレジスタの集合を R_c とすると, 外部入力から $r_c \in R_c$ までの経路(正当化経路)を生成する。ただし, r_c が無い経路は正当化経路でなくテスト経路に含まれる。 r_c または外部入力から M_i の入力ポートへテストベクトルを印加し, かつその出力応答を M_i の直後のレジスタ r_o または外部出力へ伝搬するための経路(テスト経路)を生成する。 r_o から外部出力へ出力応答を伝搬するための経路(伝搬経路)を生成する。ただし, r_o が無い経路は伝搬経路でなくテスト経路に含まれる。観測経路上に演算モジュール M_p が存在し, 観測経路が M_p の非伝搬入力を含み, M_p の伝搬入力の入出力ポート間にスルー機能が付加されていない場合は, 補助経路の生成を行う。ここで, 補助経路は外部入力から2つの入力ポートを持つ M_p の伝搬入力までの経路をいう。

図6の mux1 に対する主経路集合を図7に示す。これらの経路上の M_c を決定し, 各フェーズにおける制御ベクトルを求めることでデータバス上の回路要素全てに対してテストプランが生成できる。

テストプラン生成例 図6に示す固定制御可検査データバス中の mux1 に対するテストプラン生成を考える。 mux1 に対する制御経路は, P11 から mux1 の2つの入力ポート (x, y) までの経路である。また観測経路は mux1 の出力ポート z から PO1 までの経路である。また, $R_c = \{r_2, r_4\}$, $r_o = r_1$ である。これにより, 正当化/テスト/伝搬フェーズの経路および補助経路中に含まれる制御ポートを持つ回路要素の集合 M_c が求められる。各フェーズにおける経路中の M_c は, 正当化経路では $\{mux4, mux2, r_2\}$, テスト経路では $\{TM1, r_1\}$, 伝搬経路には, M_c は空である。以上より, これらの回路要素に対して, 各フェーズ

ごとに経路を活性化する制御ベクトルが決定する。図6の mux1 に対するテストプランを図8に示す。

3.4 テストプラン生成回路

テストプランはデータバス中の全ての組合せ回路要素に対して生成される。テストプランのデータバスへの供給は, 制御信号線を用いて行う。一般にコントローラの外部入力ビット幅は, 制御信号線のビット幅より小さいので, コントローラの外部入力から直接テストプランを供給することはできない。そのため回路内部でテストプランを生成することを提案する。コントローラ/データバス回路において全てのテストプランはテストコントローラによって生成する(図2)。テストコントローラはテストプラン生成回路 TPG , テストバタンレジスタ TPR およびテスト対象モジュールレジスタ TMR から構成される。 TMR は, テスト対象となるモジュール番号を格納する。 TMR のビット幅は, データバス中に存在する組合せ回路要素数を m とすると, \log_m である。 TPG は, TMR に設定されたテスト対象モジュール M に対応するテストプランを生成する。 TPG は組合せ回路要素 M が制御ポートを持つ場合, テストフェーズにおいて, TPR にあらかじめ格納されたテストベクトルを制御ポートに対して印加する。

表2は, テストコントローラの制御を示す。表2において t_1 は, コントローラ/データバスの通常動作およびデータバスのテストを制御するために利用する。 $mode$ は, コントローラの外部入力に接続する。これは TPR および TMR のリセット ($mode = '00'$), M に対するテストバタンの正当化 ($mode = '01'$)/テスト ($mode = '10'$)/出力応答の伝搬 ($mode = '11'$) の4制御を行うために利用するので, $mode$ のビット幅は2ビットとなる。これにより, M に対するテストプランの供給を行う。

TPG は回路の通常時のシステムクロックで動作可能なので, データバスの各組合せ回路要素に対して, at-speed テストが可能である。

4 実験結果

完全スキャン設計法, 文献[7](以後, 強可検査と呼ぶ)および本稿での提案手法を, テスト容易化設計後の面積オーバヘッド, テスト生成時間およびテスト実行時間について比較した。実験に使用した RT レベルベンチマーク回路は, GCD と PAULIN[5], 3rd lattice Wave Filter(LWF) と 4th jaumann Wave Filter(JWF)[8] である。また実設計回路として, 小規模の RISC プロセッサ(以下, RISC と呼ぶ)を使用した。これらのコントローラ/データバス回路の回路特性を表3に示す。「circuit」は回路名を示し, 「Area(#gate)」はテスト容易化設計前のコントローラ/データバス回路の面積を表す。「#PI」, 「#PO」はコントローラおよびデータバスそれぞれの外部入力数および外部出力数を表す。コントローラに関して, 「#State」, 「#Status」および「#Control」はそれぞれ, 状態数, ステータス入力数および制御出力数を表す。データバスに関して「#bit」, 「#Reg」および「#Mod」はそれぞれ, ビット幅, レジスタ数および演算モジュール数を表す。

実験に用いた論理合成ツールは AutoLogicII(Mentor Graphics), テスト生成ツールは TestGen(Synopsys), 使用した計算機は Ultra60(Sun Microsystems) である。

テスト容易化設計に伴う付加回路による面積オーバヘッ

表 4: 面積オーバーヘッド (%)

circuit	コントローラ	データバス		テストプラン生成回路		MUX.etc.		Total		
		強可検査	提案手法	強可検査	提案手法	強可検査	提案手法	完全スキャン	強可検査	提案手法
GCD	9.6	3.0	3.0	18.8	12.2	17.2	16.8	26.6	39.7	32.8
JWF	12.2	5.4	9.6	20.4	20.7	11.1	11.6	26.7	37.1	41.9
LWF	28.1	18.7	18.7	16.8	13.2	12.9	12.5	33.4	48.2	44.6
PAULIN	39.4	1.2	2.5	4.9	2.6	1.8	1.9	7.4	8.1	7.1
RISC	1.6	11.6	10.7	12.3	2.9	3.9	3.9	16.7	27.2	16.9

表 5: テスト生成結果

circuit	テスト生成時間 (sec.)			テスト実行時間 (cyc.)			故障検出効率 (%)		
	完全スキャン	強可検査	提案手法	完全スキャン	強可検査	提案手法	完全スキャン	強可検査	提案手法
GCD	171.51	0.69	0.69	6629	504	504	100.00	100.00	100.00
JWF	2.88	0.37	0.53	20519	1497	1621	100.00	100.00	100.00
LWF	0.47	0.27	0.27	4066	517	443	100.00	100.00	100.00
PAULIN	4.68	2.11	2.20	16187	2193	2172	100.00	100.00	100.00
RISC	51740.92	71.50	72.29	1006154	9674	10825	99.97	99.99	99.99

ド (単位:%) を表 4 に示す。「テストプラン生成回路」および「MUX.etc.」の面積オーバーヘッドは、テスト容易化設計前の回路面積に対する割合である。「MUX.etc.」は、MUX、テストコントローラ中の TPR および TMR を表す。ベンチマーク回路に対する全体の面積オーバーヘッドにおいて、提案手法は強可検査に比べて JWF 以外は小さく、完全スキャン設計に比べてやや大きい、差は大きくない。特に RISC に対して、提案手法は強可検査に比べて約 10% の面積オーバーヘッドを削減しており、完全スキャン設計とはほぼ同じである。TPG において、提案手法は強可検査に比べて JWF 以外は小さく、特に大規模な回路である PAULIN および RISC に対しては、TPG の面積オーバーヘッドを大幅に削減できている。

各手法のテスト生成時間 (単位:秒) を表 5 に示す。ベンチマーク回路に対する強可検査および提案手法でのテスト生成時間は、完全スキャン設計に比べて短い。特に RISC に対しては、強可検査および提案手法でのテスト生成時間は、完全スキャン設計に比べて約 1/700 と大幅に短縮している。強可検査および提案手法では、回路中の各組合せ回路要素に対してテスト生成を行うのに対して、完全スキャン設計では組合せ回路全体に対してテスト生成を行う。従って、より大規模な回路に対しては、強可検査および提案手法では、完全スキャン設計よりも大幅にテスト生成時間の短縮が期待できる。

各手法のテスト実行時間 (単位:クロックサイクル) を表 5 に示す。完全スキャン設計では、テスト実行時間を「テストベクトル数」×(「FF 数」+1)+「FF 数」とした。強可検査および提案手法では、テスト実行時間を全ての組合せ回路要素の「テストベクトル数」×「テストプラン長」の和とした。強可検査および提案手法のテスト実行時間は、ベンチマーク回路では完全スキャン設計に比べて約 1/10 と短く、特に RISC では、完全スキャン設計に比べて約 1/100 と大幅に短縮している。

各手法の故障検出効率 (単位:%) を表 5 に示す。強可検査および提案手法では、RISC に対する故障検出効率は 99.99% である。これは、組合せ回路用テスト生成アルゴリズムツールが RISC に含まれる乗算器に対してテストパターンを一部生成できないためである。ベンチマーク回路に対する故障検出効率は全て 100% を達成している。

5 むすび

本稿では、RT レベルでのコントローラ/データバス回路に対するテスト容易化設計法を提案した。提案手法では、データバスのテスト容易性として新たに固定制御可検査性を定義し、固定制御可検査性に基づくテスト容易化設計法、およびテストプラン生成法を提案した。データバスのテスト生成法は、階層テスト生成法に基づいており、各組合せ回路要素に対して組合せ回路用のテスト

生成アルゴリズムを適用できるので、短いテスト生成時間で完全故障検出効率を保証している。また固定制御可検査性は、データバス中の各組合せ回路要素のテストプランの制御ベクトル数は高々 3 であり、TPG は組合せ回路で構成できる。従って、提案手法では TPG の面積は文献 [7] に比べて小さくなる。ベンチマーク回路および実設計回路を用いた実験では、提案手法は文献 [7] と比較して、テスト生成時間、テスト実行時間の点で同等で、面積オーバーヘッドは小さい。

謝辞 本研究に際し、多くの貴重な意見を頂いた奈良先端科学技術大学院大学の増澤利光助教授、井上美智子助手ならびに広島市立大学の井上智生助教授に深く感謝します。本研究は一部、(株)半導体理工学研究センター (STARC) との共同研究、及び、文部省科学技術研究費補助金・基盤研究 B(2) (課題番号 09480054) の研究助成による。

参考文献

- [1] H.Fujiwara, *Logic Testing and Design for Testability*, The MIT press, 1985.
- [2] P.C.Maxwell, R.C.Aitken, V.Johansen, and I.Chiang, "The effect of different test sets on quality level prediction: when is 80% better than 90%?," in *Proc. of Int. Test Conf.*, pp.358-364, 1991.
- [3] 大竹哲史, 増澤利光, 藤原秀雄, "完全故障検出効率を保証するコントローラ非スキャンテスト容易化設計法", 電子情報通信学会論文誌, Vol.J81-D-I, No.12, pp.1259-1270, Dec. 1998.
- [4] B.T. Murray and J.H. Hayes, "Hierarchical test generation using pre computed tests for modules," *IEEE Trans. on CAD*, VOL.9, NO.6, pp.594-603, June 1990.
- [5] I.Ghosh, A.Ragunathan, and N.K.Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," *IEEE Trans. on CAD*, VOL.16, NO.9, pp.1001-1014, Sep. 1997.
- [6] 和田弘樹, K.K.Saluja, 増澤利光, 藤原秀雄, "完全故障検出効率を保証するレジスタ転送レベルデータバスの非スキャンテスト容易化設計法", 電子情報通信学会論文誌, Vol.J82-D-I, No.7, pp.843-851, Jul. 1999.
- [7] S.Ohtake, H.Wada, T.Masuzawa and H.Fujiwara, "A non-scan DFT method at register-transfer level to achieve complete fault efficiency," in *Proc. of ASP-DAC*, 2000 (掲載予定)
- [8] M.Inoue, T.Higashimura, K.Noda, T.Masuzawa, and H.Fujiwara, "A high-level synthesis method for weakly testable data paths," in *Proc. IEEE the 7th Asian Test Symposium*, pp.40-45, 1998.